How to export large data volumes to CSV using Mashups in Qlik Sense

Version: 1.0

Date: 2016-11-10

Creator: Levi Turner

**Disclaimer:**

**Please be aware that this document is not supported and is meant only as a guide.  Individual environments may require adjustments for things to work correctly.This will/may require altering and customizing code shipped with QlikView Server/Sense Enterprise. All customization is done at your own risk and is not covered by Qlik Support or Maintenance Agreements.  Please backup any files prior to modification.**

Table of Contents

[1 Introduction 2](#_Toc466534097)

[2 Configuring the Engine to export large data volumes 2](#_Toc466534098)

[3 Creating the Mashup to do the export 2](#_Toc466534099)

# Introduction

In Qlik Sense, exports in the Hub are truncated in order to limit the resources required for exports. In order to successfully When exporting a visualisation Object, the Engine has to locate the associative [granular] data, construct a HyperCube in RAM, calculate visible aggregates, explode the HyperCube in memory and dump it to a temporary file, notify the browser that a temporary file is available, deliver it across the network, and then let the browser write the file to disk.  This process can take unexpected resources to produce, especially with large data volumes.

The recommended approach is to rarely, if ever, use the front-end UI to export data, but instead use the STORE function to write to CSV in the load script then share with the users who need access to this raw data. But this recommendation may not fit all use cases. To achieve large data exports, it is necessary to change settings on the Engine as well as use of a Mashup to directly call the export to CSV. Other approaches may be possible, but will not be covered here.

# Configuring the Engine to export large data volumes

In order for the Engine to export large data volumes to CSV, you must first over-ride its default limitation of 5 million records. To do this

* Stop the Qlik Sense Engine and Qlik Sense Service Dispatcher services.
* Open Notepad, or your preferred text editor as administrator
* Make a copy of the Engine’s Settings.ini, which is located by default in C:\ProgramData\Qlik\Sense\Engine\Settings.ini
* Open the Engine’s Settings.ini in Notepad
* Edit the file to have the following elements:
	1. [Settings 7]: This will tell the Engine to look for non-default settings
	2. ExportMaxCellsCsv=VALUE, where VALUE is a number that is larger than 5,000,000
	3. A blank link at the end
* The resulting file will look similar to this (I am using the Sublime text editor in this example):



* Start the Qlik Sense Engine and Qlik Sense Service Dispatcher services

# Creating the Mashup to do the export

To create the mashup, you can use the sample extension which is included in this article (ExportCSV.zip) as a template for customization.

* Import the sample extension into the QMC > Extensions
* Go to the Hub to access the Dev-Hub



* Open the Mashup named Export Large Volumes
* Also open up the Single configurator
* In the Single Configurator, navigate to the app which has the table which you want to export and click on the table. Make a note of the App ID (f4ebe42e-a621-4d49-a976-4f721e520862 in this example) and object ID (JFNw in this example).



* In the Mashup editor, navigate to the ExportCSV.html tab and insert your App and object IDs into the code then save the changes



* Navigate to the Preview tab and the table that is present in your app should render. Now click on the Export chart 1 CSV button.



* The Engine will now generate the CSV and deliver it to your browser. Open the CSV in the utility of your choosing and validate that it contains the expected number of rows. You may need to use a specialized tool since the resulting file can be quite large. For reference, a table with 20,000,000 rows produced a CSV which was 180MB.

# Considerations

This approach should be used judiciously and extensive testing at scale should be done to ensure that this change of behaviour will not negatively impact the end user experience in the Hub / integration.